Partioning Nearest Neighbour approach to Regression Variation Improvement in tree-based Approaches

Abhinav Mathur1, Sunny Verma2

1 – Clinton Health Access Initiative, 2 – Absolut Data Research & Analytics Private Ltd

# Abstract

Good generalized machine learning models should have high variability post learning 1. Tree-based approaches2 are very popular due to their inherent ability in being visually representable for decision consumption as well as robustness and reduced training times. However, tree-based approaches lack the ability to generate variations in regression problems. The maximum variation generated by any single tree-based model is limited to the maximum number of training observations considering each observation to be a terminal node itself. Such a condition is an overfit model. This paper discusses the use of a hybrid approach of using two intuitive and explainable algorithms, CART2 and k-NN3 regression to improve the generalizations and sometimes the runtime for regression-based problems. The paper proposes first, the use of using a shallow CART algorithm (Tree depth lesser than optimal depth post pruning). Following the initial CART, a KNN Regression is performed at the terminal node to which the observation for prediction generation belongs to. This leads to a better variation as well as more accurate prediction than by just the use of a CART or a KNN regressor as well as another level of depth over an OLS regression1. The paper shall present the algorithm referred to as PNN (Partioned Nearest Neighbours) and it’s benefits when dealing with higher dimension data

# Introduction to CART & KNN algorithm

## CART (Classification & Regression Trees)

CART stands for **C**lassification **A**nd **R**egression **T**rees and among one of the most popular and widely used types of tree based algorithms. CART uses the concept of information gain/gini with respect to the bins of the target variable to partition data into smaller buckets. The numeric value given is the mean of the target variable in the data points in the terminal bucket, also called as the leaf node. Another useful feature of CART is taking a bivariate approach in portioning, i.e. – For each split or partition performed by the algorithm, it will only add a condition to one variable. This allows us to also reconstruct variable importance

Since partitioning of data can be reconstructed as a top down approach with the variable and level explaining the highest amount of gini/information gain, a hierarchy of the splits also provides an insight into variable importance.

A sequence of data partitions can be visualized as a set of decision rules in the form of a tree of if-else statements leading to the popular term of ‘decision trees’(Reference needed).

## K-NN (K- Nearest Neighbours regression)

K-NN is a lazy evaluation algorithm that finds ‘K’ most similar observations in terms of mathematical distances in an N-dimensional space where N is the number of the independent variables used to predict the dependent variable. Two of the popular distance measures used are Eucilidean, Manhattan distances explained as below.

A K-NN calculates distances for a cartesian product between all train and test data observations and returns the mathematical average of the dependent variable of the ‘K’ similar observations. Since K-NN computes the distance between each possible pair of observations in the training and test data sets, it is again an intuitive algorithm which can be understood in terms of the most possible value arising between similar observations.

# Drawbacks of CART & KNN algorithm

While CART and K-NN are easily intuitive algorithms, they suffer from some critical computational and predictive lapses on their own. Some of the most common lapses in these algorithms can be summarized in the following points

## Drawbacks of CART

1. CART does not give good generalizations for regressions as the number of possible regression values generated from CART are in the superset of the distinct regression values present in the training dataset
2. CART is susceptible to missing out relationship type as the splits for numeric values are calculated by sorting of values and then choosing a mid-point, this could result in a split that misses out the actual relationship value. This becomes more prominent in sparse datasets.

## Drawbacks of K-NN

1. K-NN algorithms are very computationally expensive as they calculated an N dimensional distance over a cartesian product between the training and test datasets
2. Even with space portioning techniques (kd-tree), bisecting clusters etc, the space reduction happens only by the scale of distance, this could result in the loss of important values with respect to target variable. This again becomes more prominent in sparse datasets.
3. K-NN algorithms run time increases at an exponential rate based on the increase in test / prediction data observations

# Proposed Algorithm

The algorithm proposed in this paper, hence referred to as Partioned Nearest Neighbours (PNN) throughout the course of the paper is an ensemble of a weak learner (i.e. a shallow decision tree) along with a knn run on the terminal nodes for the test observations which leads to a reduced search space along with the scope of multi-threading the process for higher core machines.

A CART algorithm is first run on the training dataset with a reduced depth (less depth than a pruned model or arbitrarily chosen by the Data Scientist) which would generate partition rules based on Information gain with respect to the target variable. A shallow CART will allow to find the most important variables with the highest predictive power and hence

The rules can be applied to get the partition index or the terminal node for the test data observations. For each test data point in a partition index, the algorithm will seek to run to find its ‘K’ nearest neighbours algorithmically with Euclidean distance. The K shall be automated to the following scenario.

1. If training samples in the terminal node are less than 10, then K = 3
2. If training samples in the terminal node are greater than 10, then K = 5 or K = 2% of N, whichever is lower. N is the number of training samples in the terminal node

The K-NN regression returns the values of the arithmetic mean of the values of the nearest neighbours as the prediction for the observation

## Flowchart

## Math

# Performance

# Multi-Threading Possibility

# References

1 – Bengio, Y., 2009. Learning deep architectures for AI. Foundations and trends® in Machine Learning, 2(1), pp.1-127.

2 – Li, B., Friedman, J., Olshen, R.A. and Stone, C.J., 1984. Classification and Regression Trees (CART). Encyclopedia of Ecology, 40(3), pp.582-588.

3 – Varmuza, K., 1980. K—Nearest Neighbour Classification (KNN-Method). In Pattern Recognition in Chemistry (pp. 62-71). Springer, Berlin, Heidelberg.